

Figure 1

10

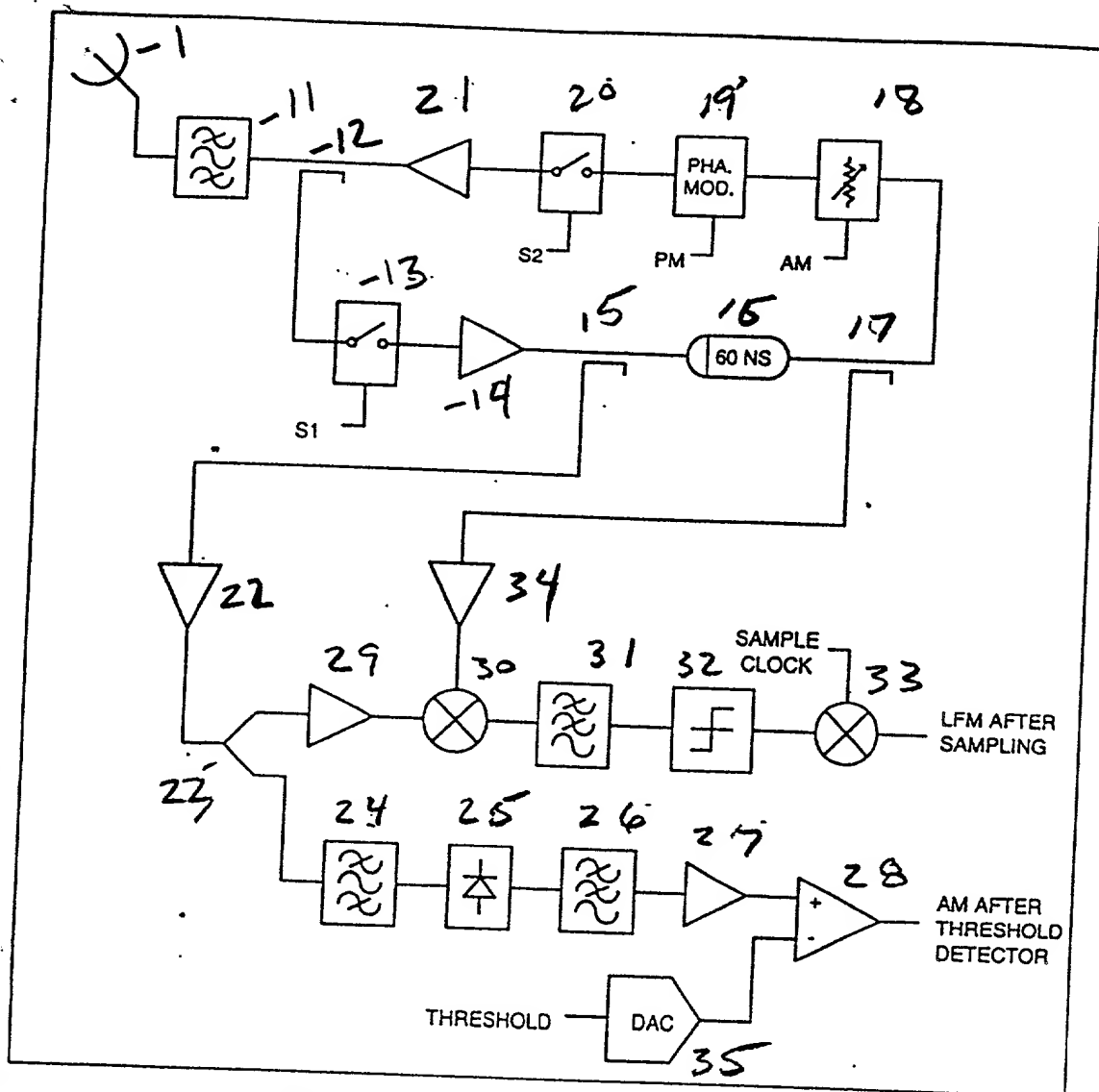


Figure 2

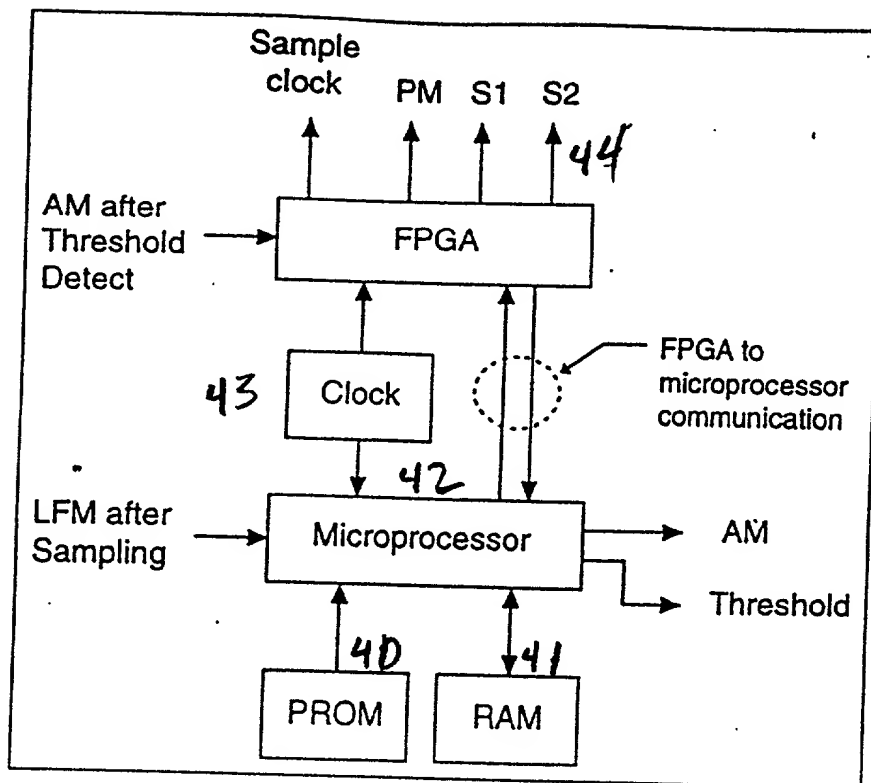


Figure 3

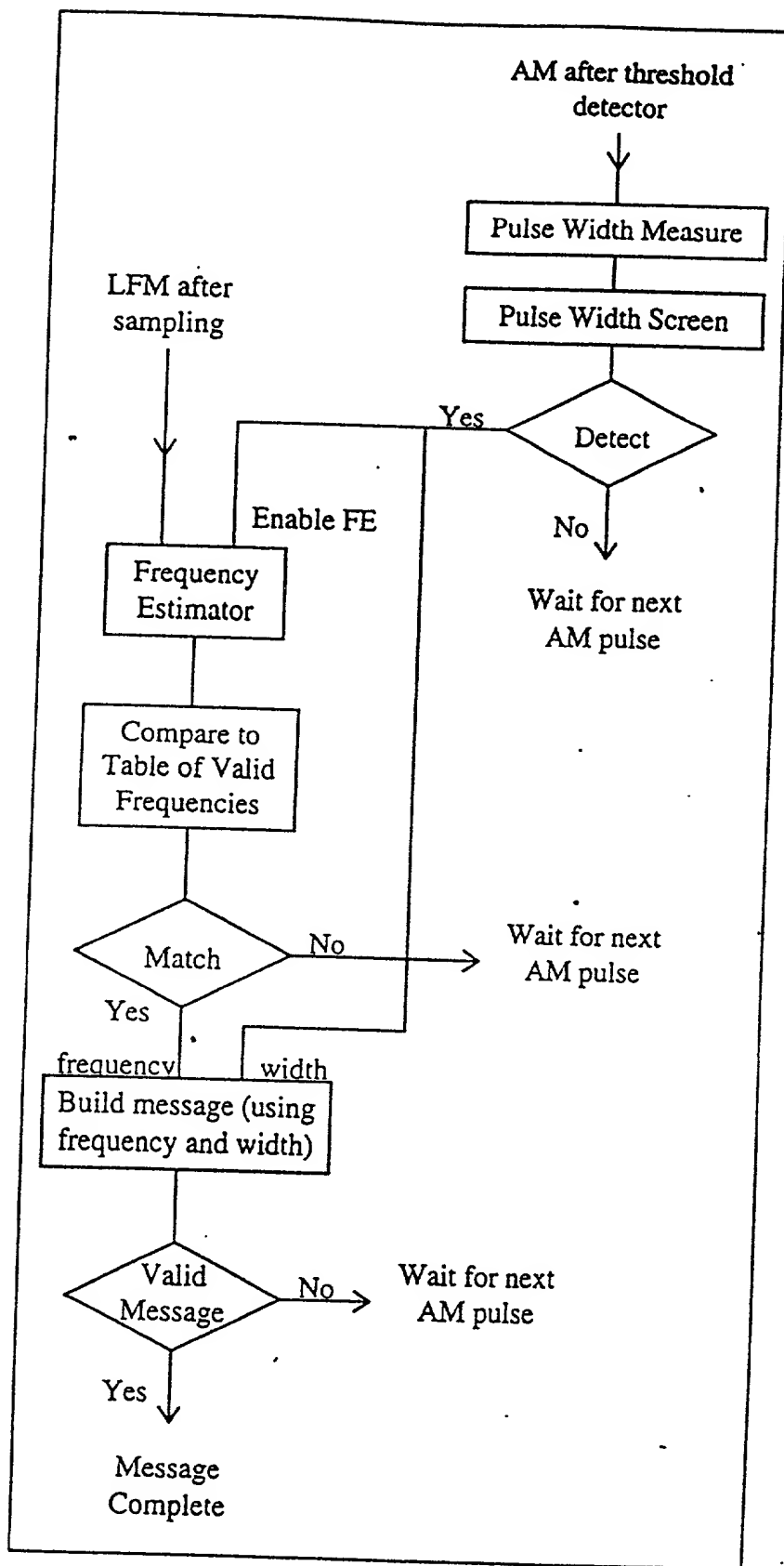


Figure 4

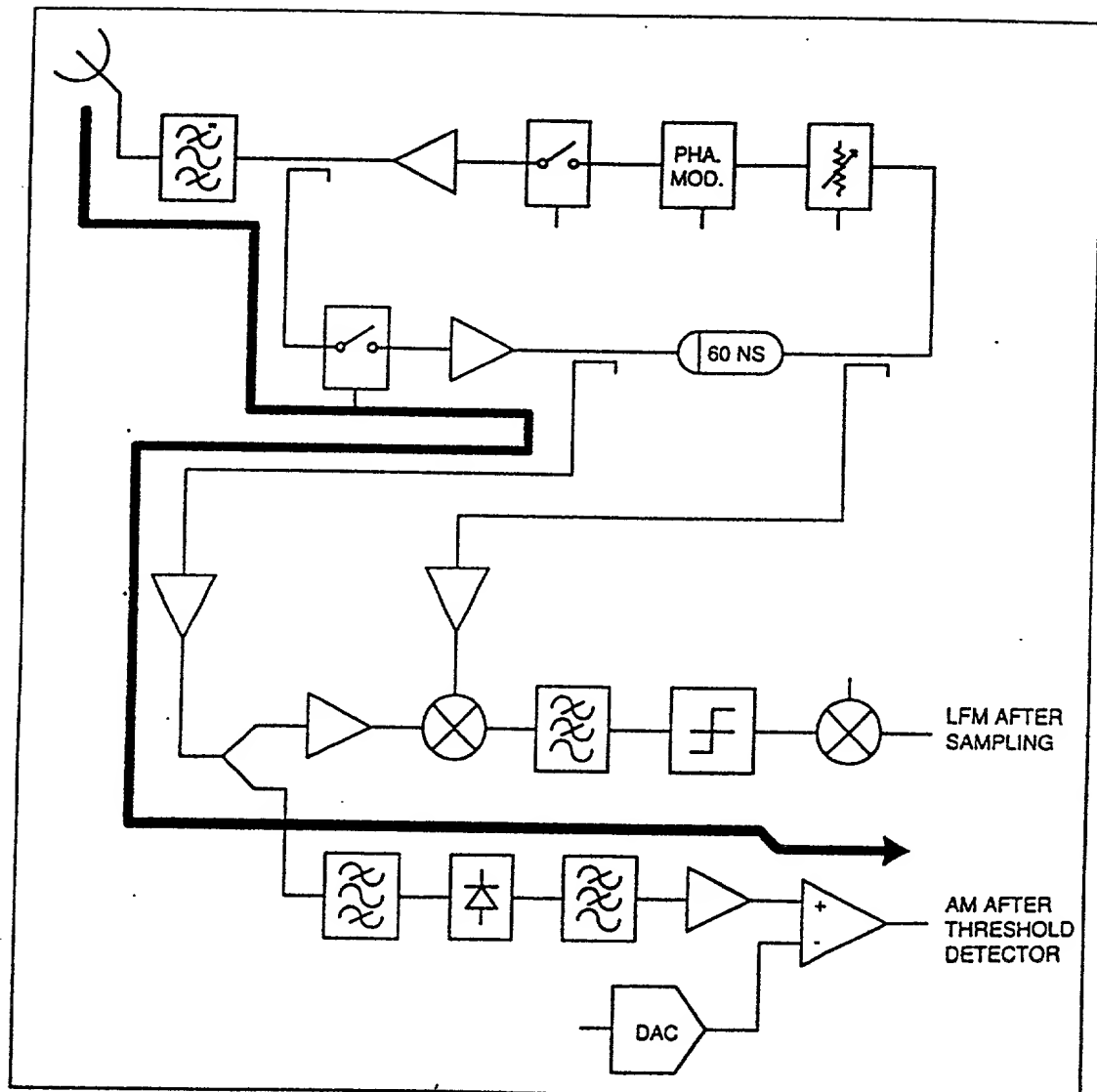


Figure 5

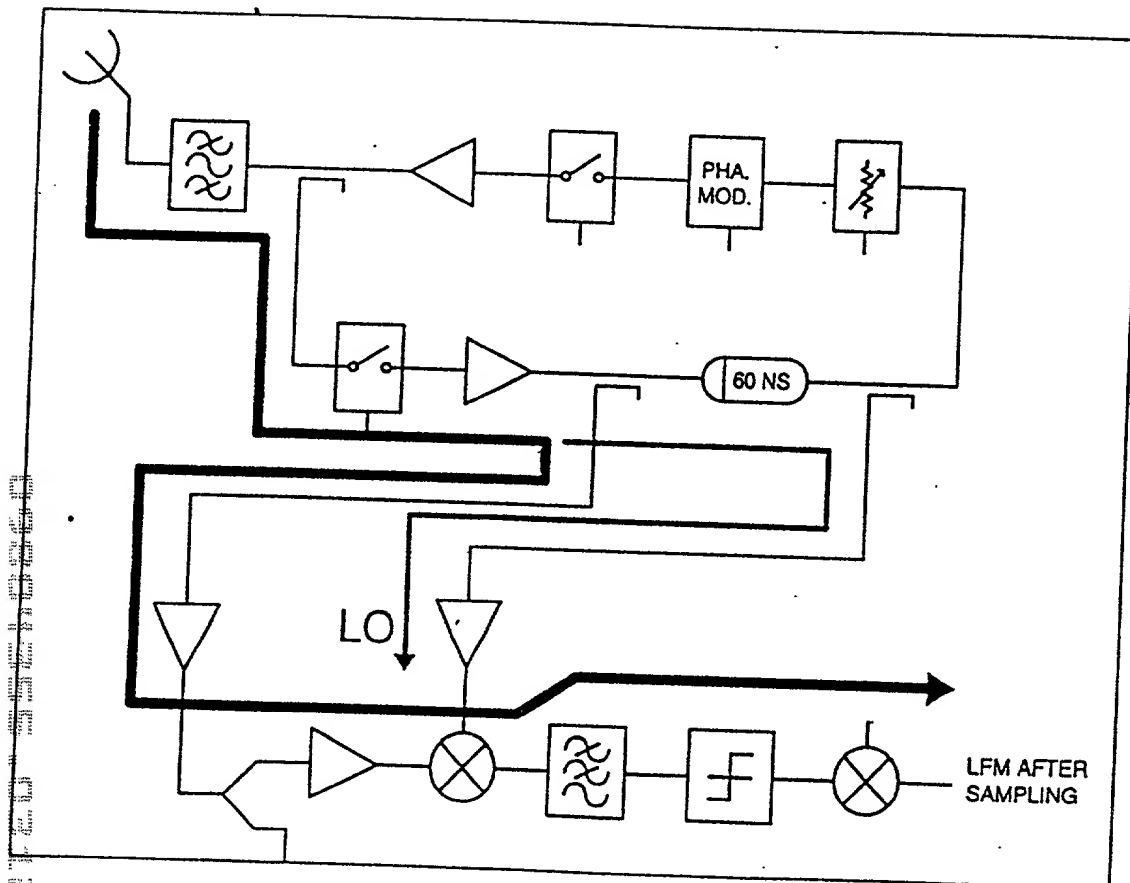


Figure 6

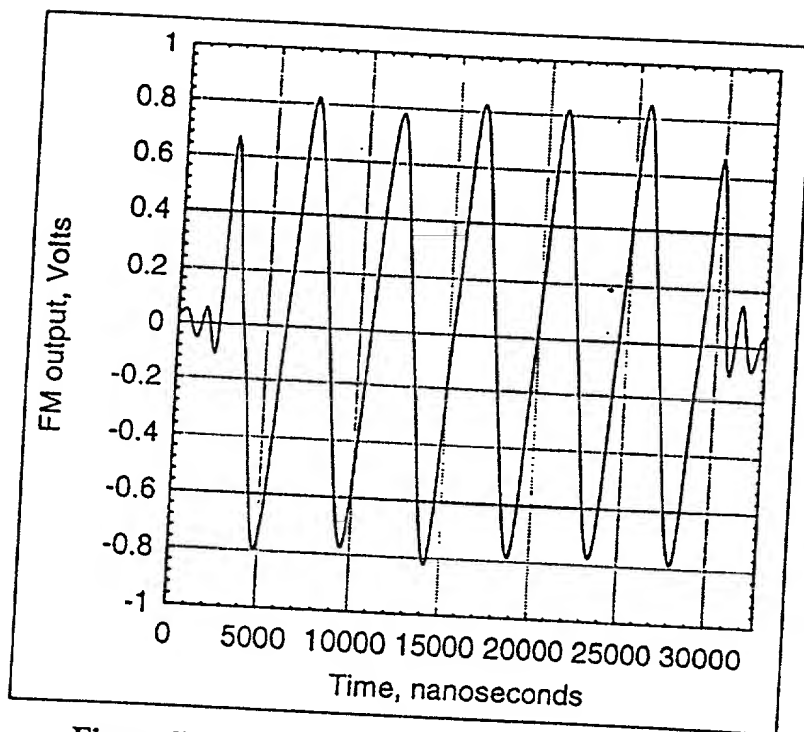


Figure 7 a

```

transition = 0
cycles = 0
cycles_temp = 0
clock = 2e6
for (n = 2 ; n <= data ; n++)
{
    if ((SS[n] == 1) && (SS[n-1] == 1))
    {
        cycles_temp = cycles_temp + 1
        cycles = cycles + 1
        if (SLIN[n-1] != SLIN[n])
        {
            transition = transition + 1
            cycles_temp = 1
        } else {
        }
        if (transition == 0)
        {
            cycles = 0
            cycles_temp = 0
        } else {
        }
    } else {
    }
}
cycles = cycles - cycles_temp
if (cycles <= 0)
{
    cycles = 1
} else {
}
frequency_estimate = (transition - 1) * clock / (cycles * 2)

```

Figure 7b

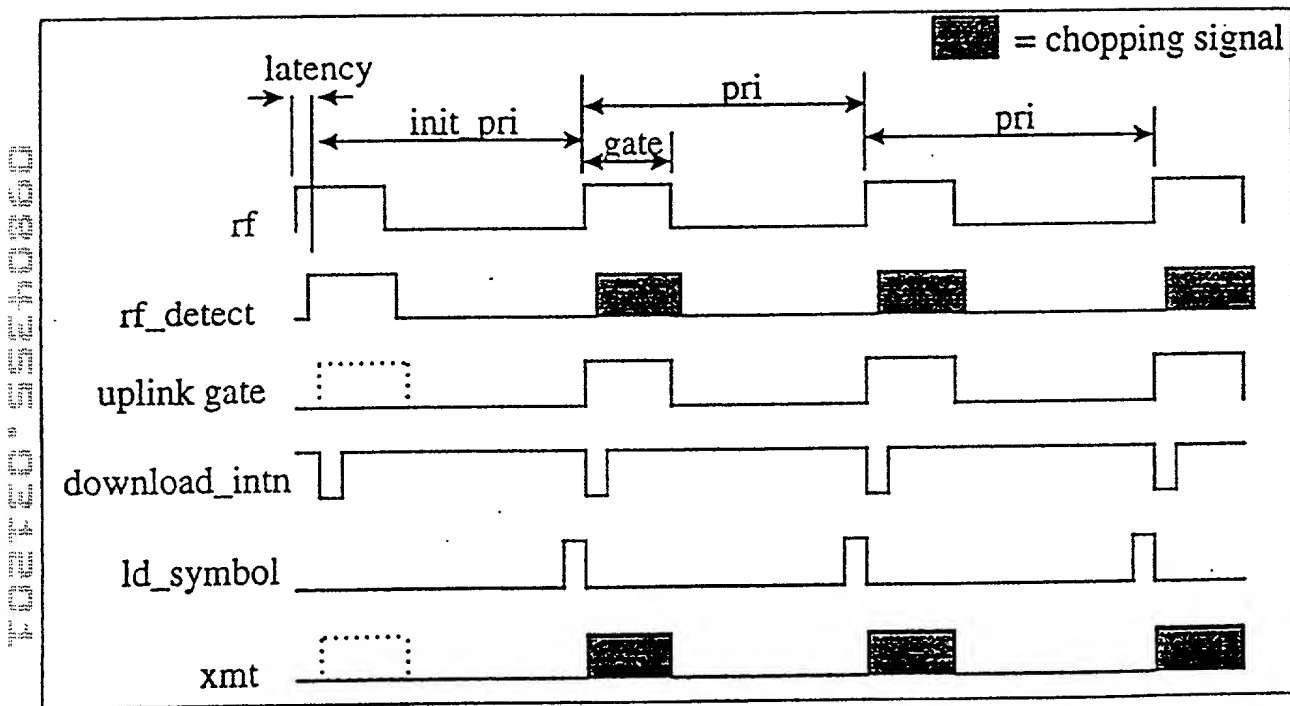


Figure 8

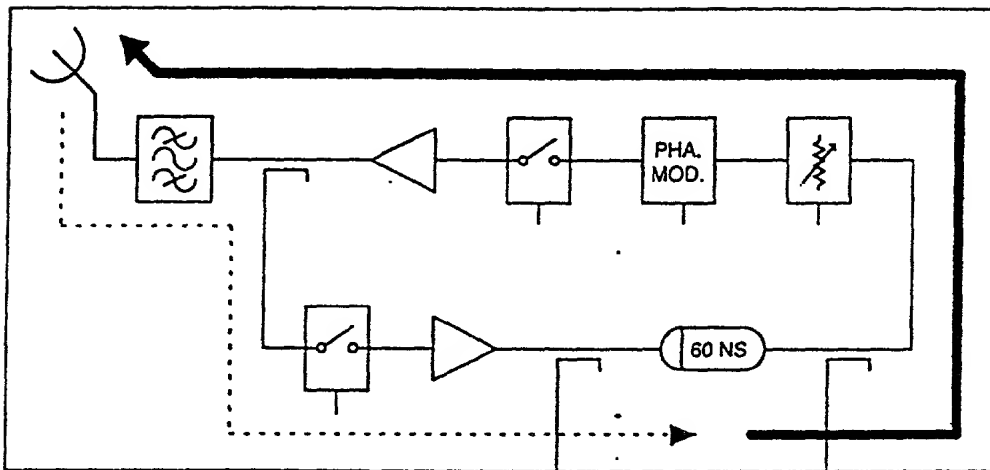


Figure 9

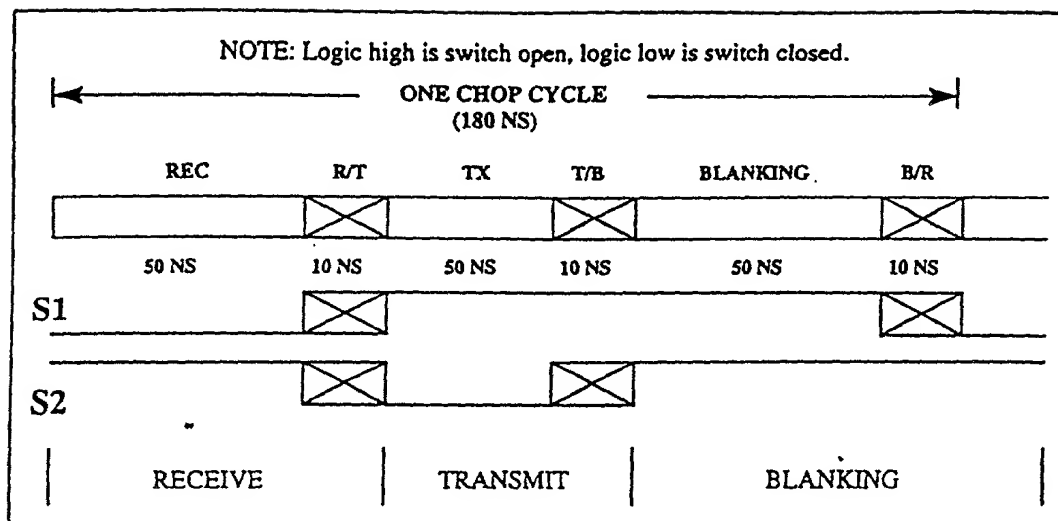


Figure 10

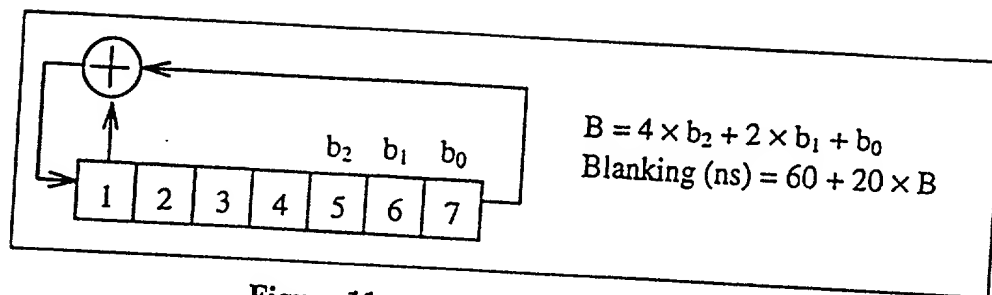


Figure 11 a

DATA BIT	AVERAGE FREQUENCY DEVIATION (Hertz)	PHASE (degrees)
000	+3100586	+90
001	+1025391	+90
010	-3100586	+90
011	-1025391	+90
100	+3100586	-90
101	+1025391	-90
110	-3100586	-90
111	-1025391	-90

Figure 11b

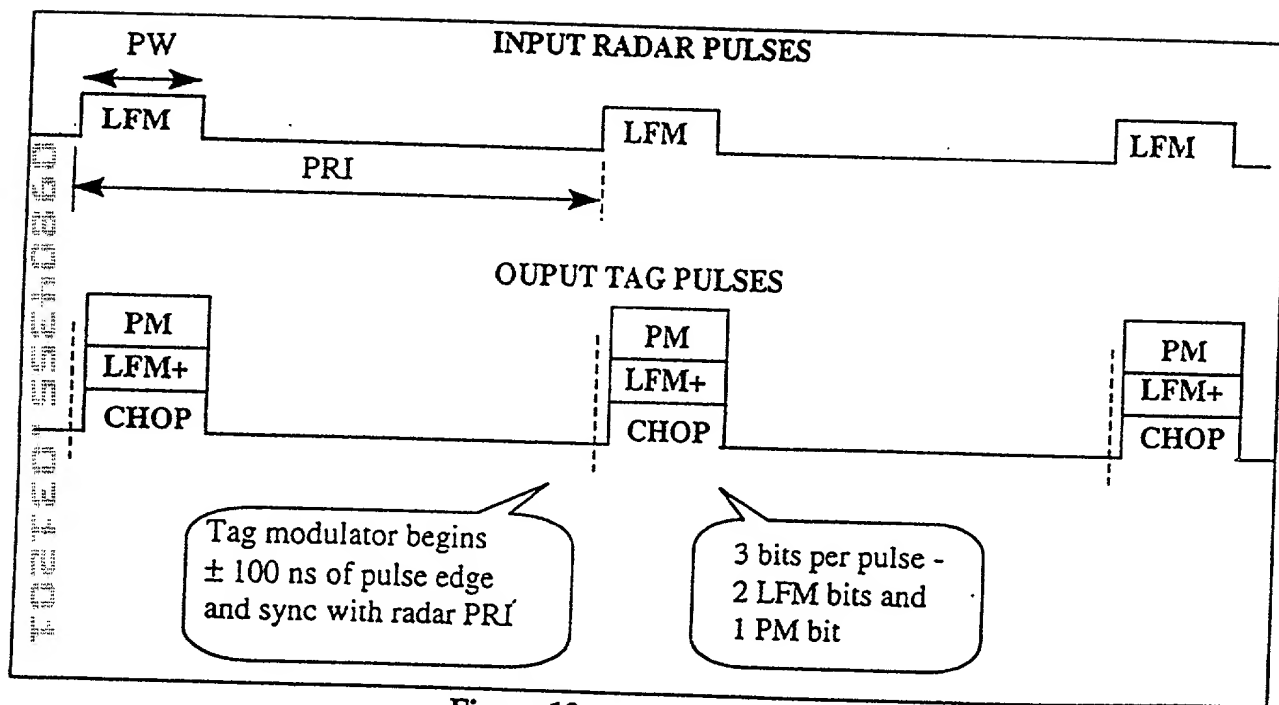


Figure 12

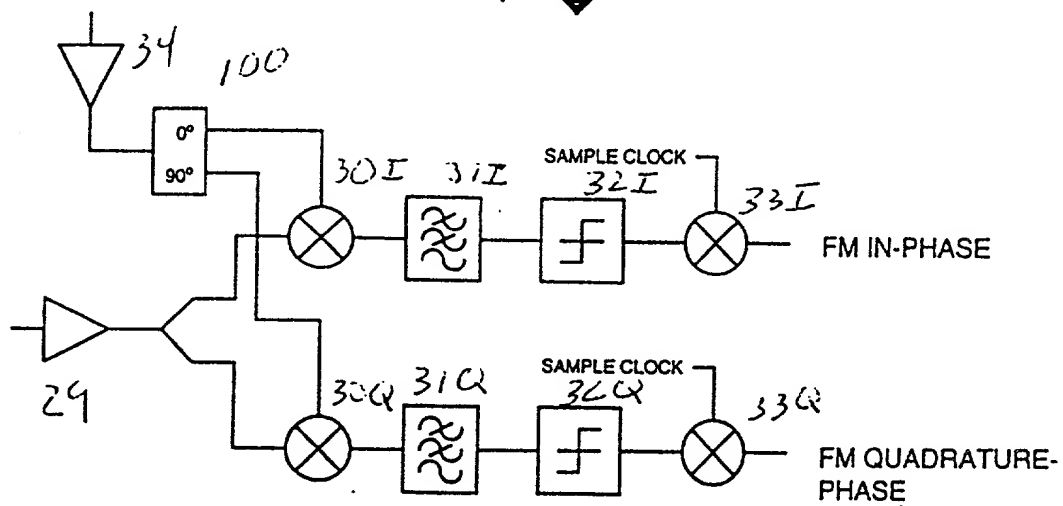
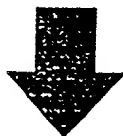
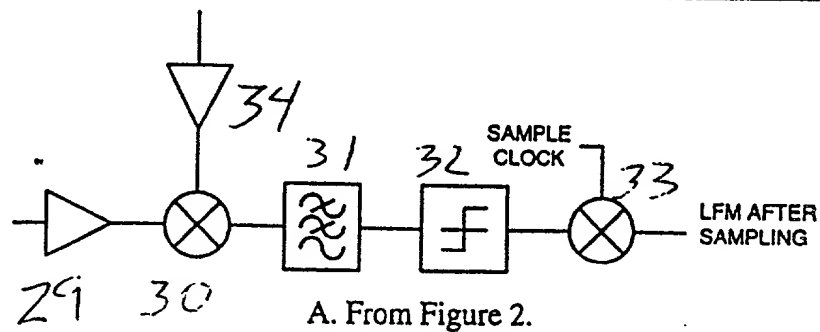


Figure 13

Radio Tag for LFM Radar

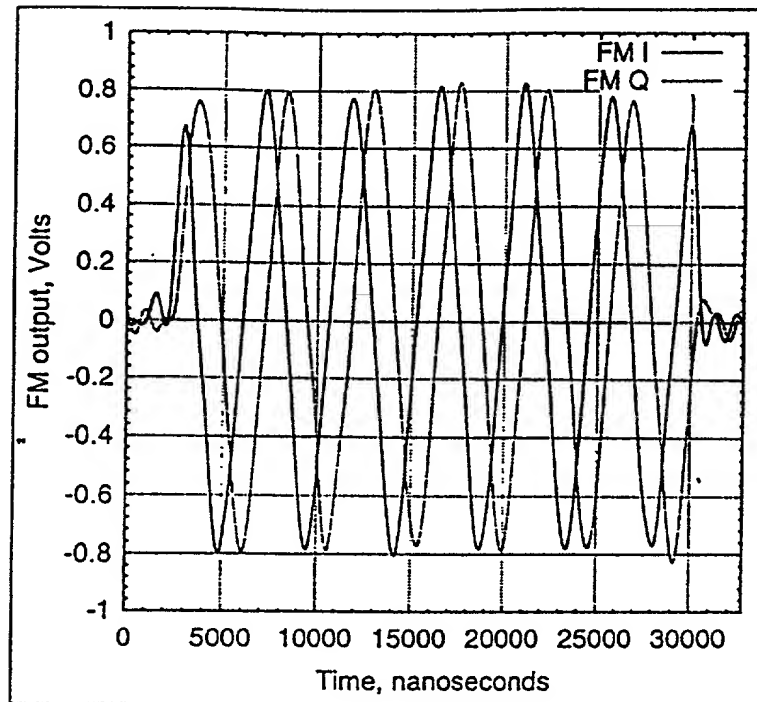


Figure 14a

```

for (n = 2 ; n <= data ; n++)
{
    if ((SS[n] == 1) && (SS[n-1] == 1))
    {
        if ((SLIN_Q[n-1] != SLIN_Q[n]) && SLIN_Q[n] == 1)
        {
            SLOPE[n] = SLIN_I[n]
        }
    }
}

```

Figure 14b

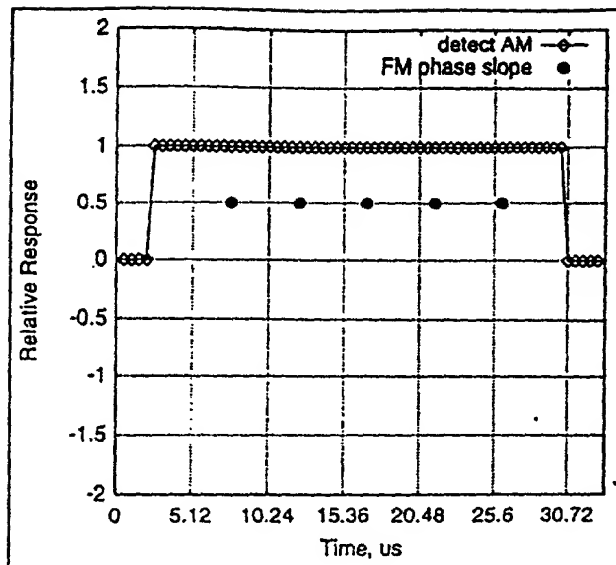


Figure 15

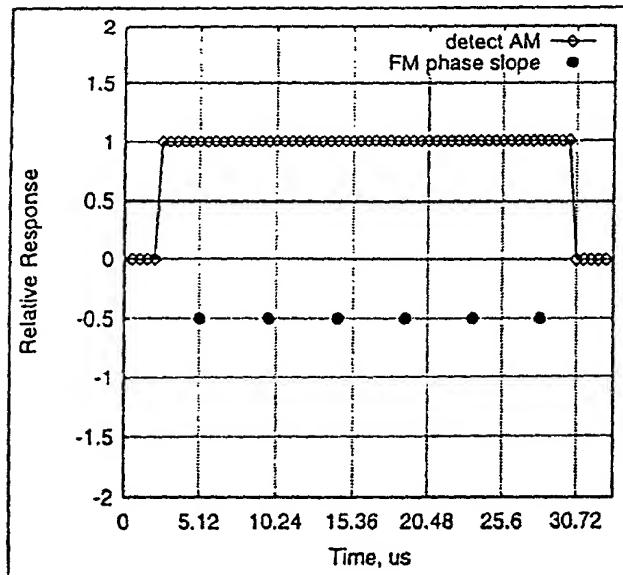


Figure 16